Fish Analysis and Regression

Introduction

This is my report of an Exploratory Data Analysis and Regression of a Fish dataset I obtained from Kaggle. We began by installing packages and loading them into R. We then do a quick summary of the data to see how the data is distributed. Since we are dealing with numerical data most of our graphs will be scatter plots to see how the variables work together. We will then create a correlation plot of our variables to get the correlation of all variables. Next, we continue our project by manually splitting the data into test and train sets then predicting our model on our test set. This was just to show how it can be done manually, but I use caret to run a cross validation model to get a more accurate assessment of our model and model error.

We began by importing some useful packages that is going to make accomplishing this project much easier. Next we are going to import our data into R from Excel using the readxl function.

The column names in this dataset are vague, so we will explain what each one means.

- Species - Species name of fish
- Weight - Weight of fish in grams
- Length1 - Vertical Length in cm
- Length2 - diagonal length in cm
- Length3 - cross length in cm
- Height - height in cm
- Width - diagonal width in cm

```
head(Fish)
```

```
## # A tibble: 6 x 7
##   Species Weight Length1 Length2 Length3 Height Width
##   <chr>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl> <dbl>
## 1 Bream      242    23.2    25.4    30     11.5  4.02
## 2 Bream      290    24      26.3    31.2   12.5  4.31
## 3 Bream      340    23.9    26.5    31.1   12.4  4.70
## 4 Bream      363    26.3    29      33.5   12.7  4.46
## 5 Bream      430    26.5    29      34     12.4  5.13
## 6 Bream      450    26.8    29.7    34.7   13.6  4.93
```

```
Fish <- Fish %>% select(-Length2)
summary(Fish)
```

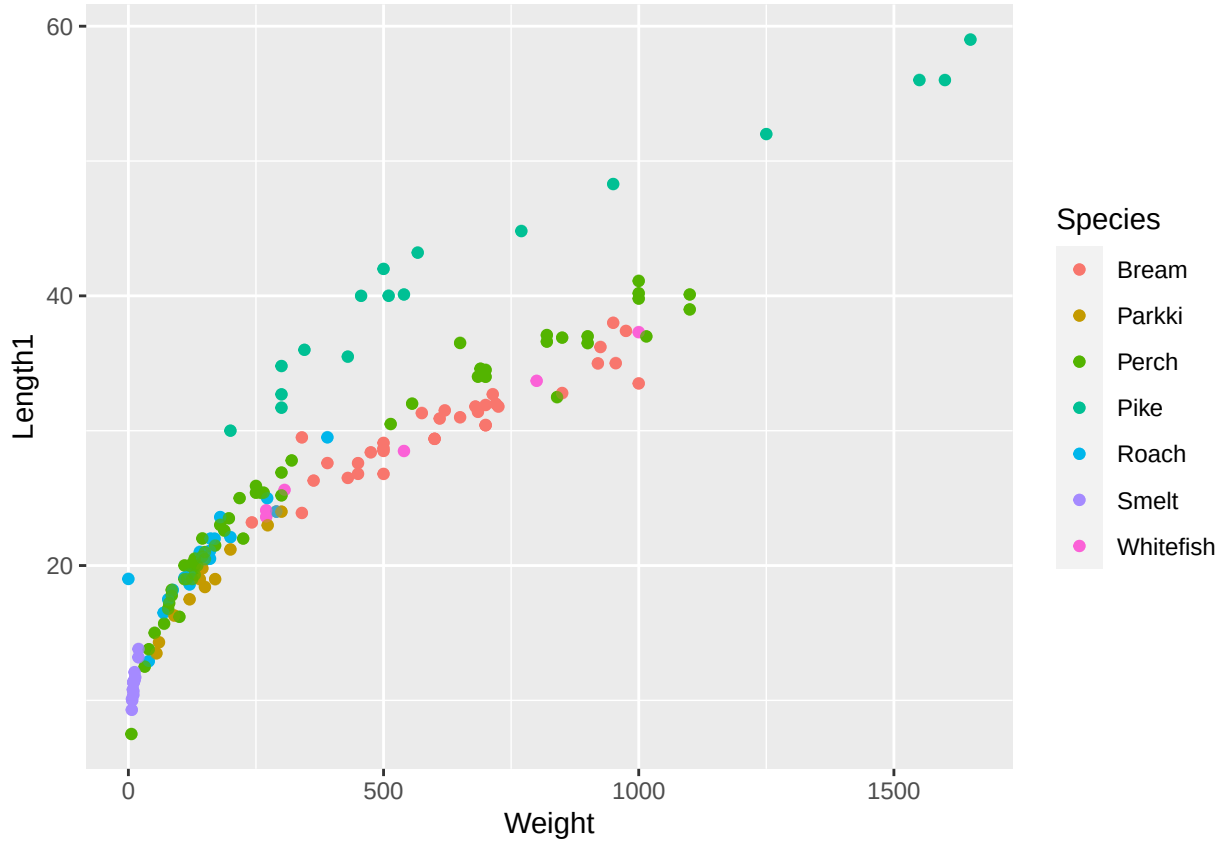
```
##    Species              Weight          Length1         Length3
##  Length:159         Min.   :   0.0   Min.   : 7.50   Min.   : 8.80
##  Class :character   1st Qu.: 120.0   1st Qu.:19.05   1st Qu.:23.15
##  Mode  :character   Median : 273.0   Median :25.20   Median :29.40
##                     Mean   : 398.3   Mean   :26.25   Mean   :31.23
##                     3rd Qu.: 650.0   3rd Qu.:32.70   3rd Qu.:39.65
```

```
##                     Max.   :1650.0   Max.   :59.00   Max.   :68.00
##      Height           Width
##  Min.   : 1.728   Min.   :1.048
##  1st Qu.: 5.945   1st Qu.:3.386
##  Median : 7.786   Median :4.248
##  Mean   : 8.971   Mean   :4.417
##  3rd Qu.:12.366   3rd Qu.:5.585
##  Max.   :18.957   Max.   :8.142
```

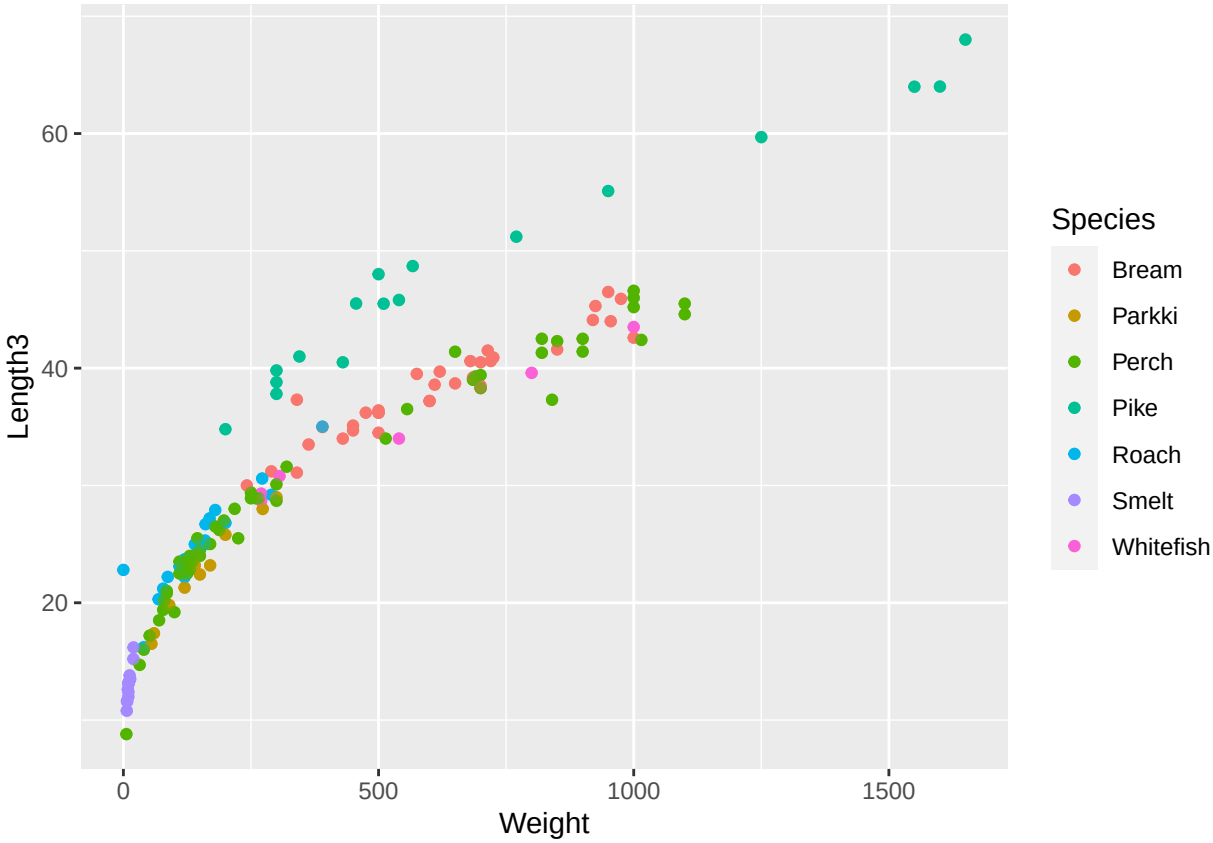
The dataset had the columns names on the first row of the actual dataset. To fix this problem we simply skip the fist line only importing the values. The diagonal length was not import to me so I remove the column from the dataset as well. Summary is a very useful function as we can quickly see min, max, and averages throughout the dataset. We see that the mean weight is 398.3 grams and mean height is 8.971 cm.

EDA

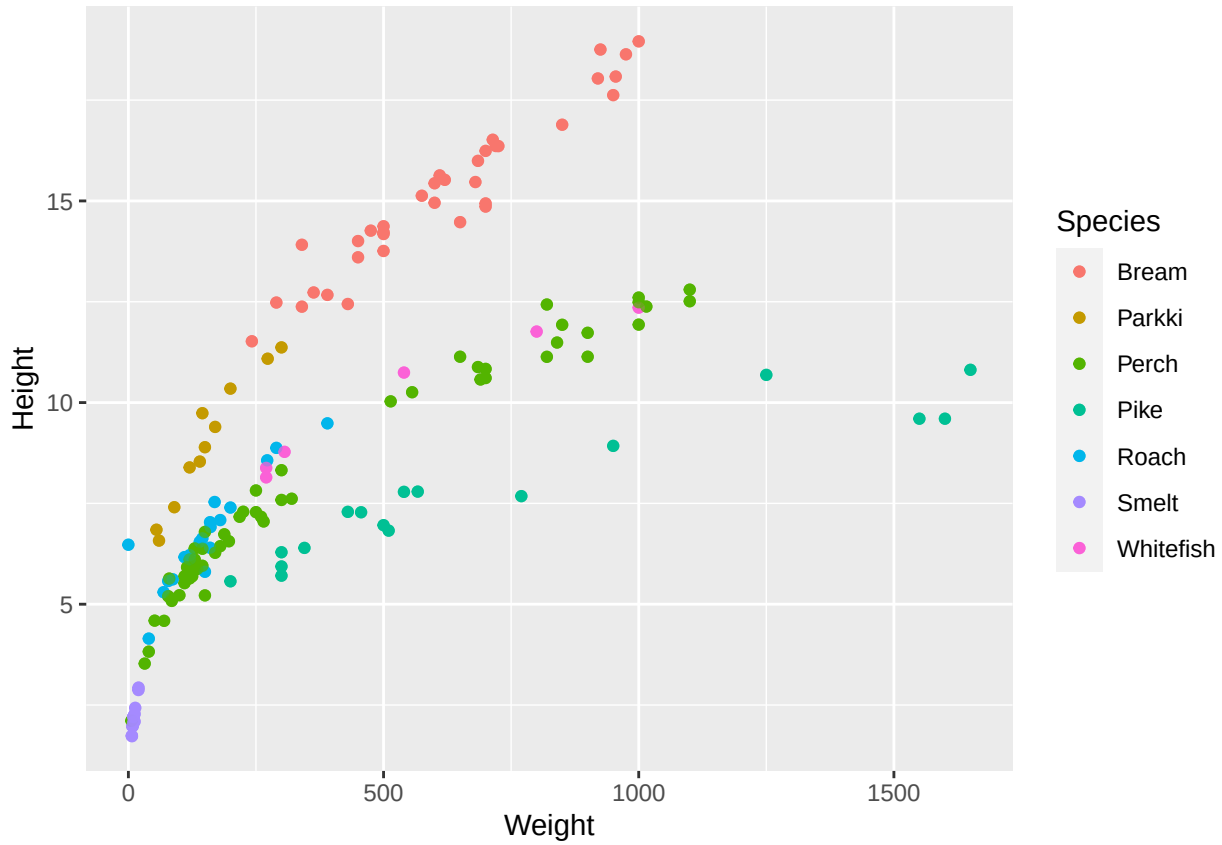
```
ggplot(Fish, aes(x = Weight, y = Length1, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```



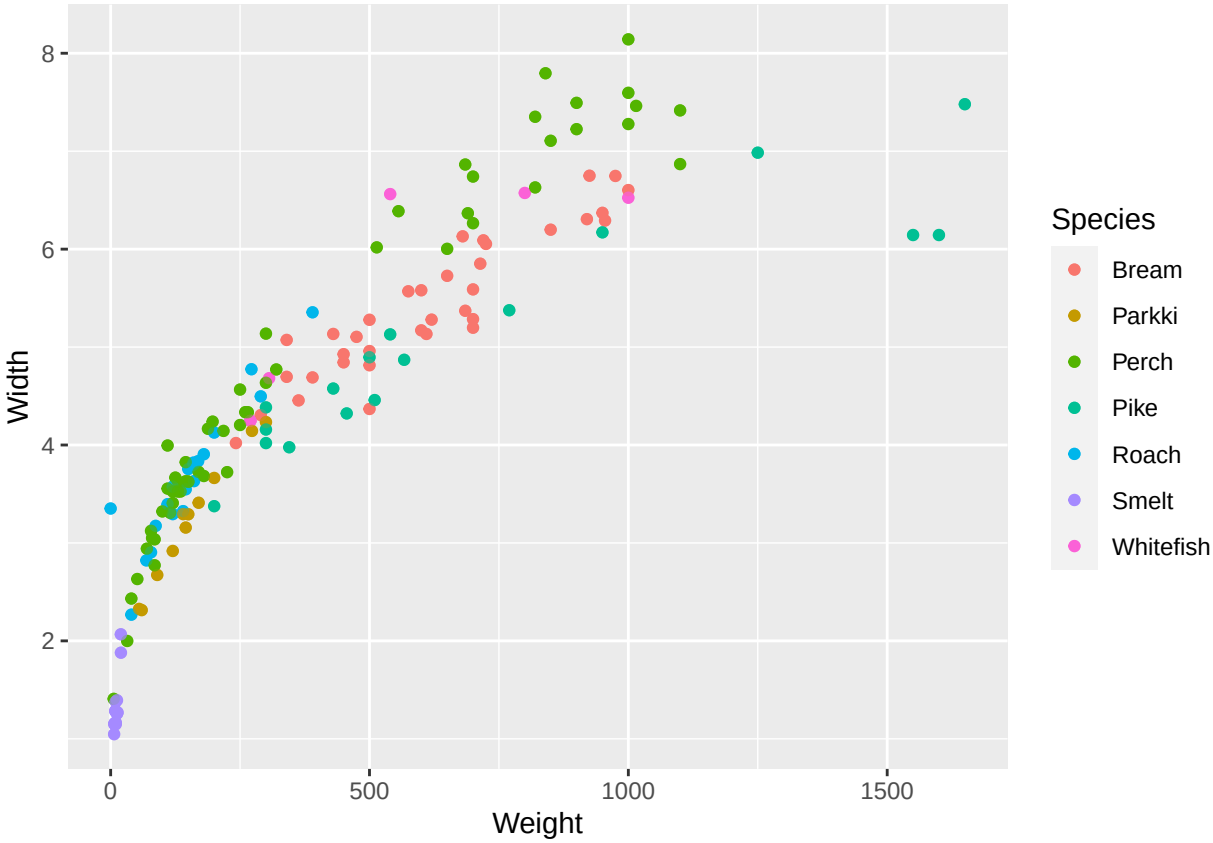

```
ggplot(Fish, aes(x = Weight, y = Length3, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```



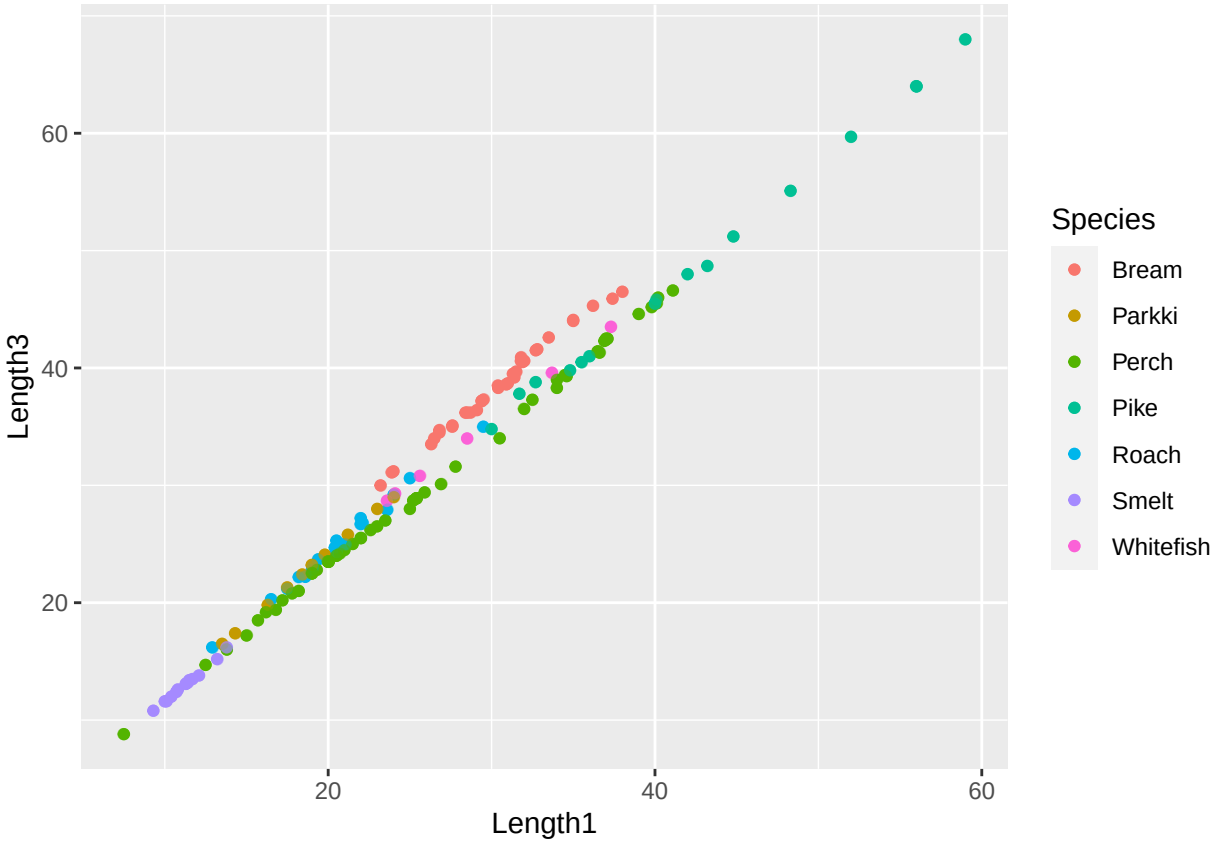

```
ggplot(Fish, aes(x = Weight, y = Height, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```



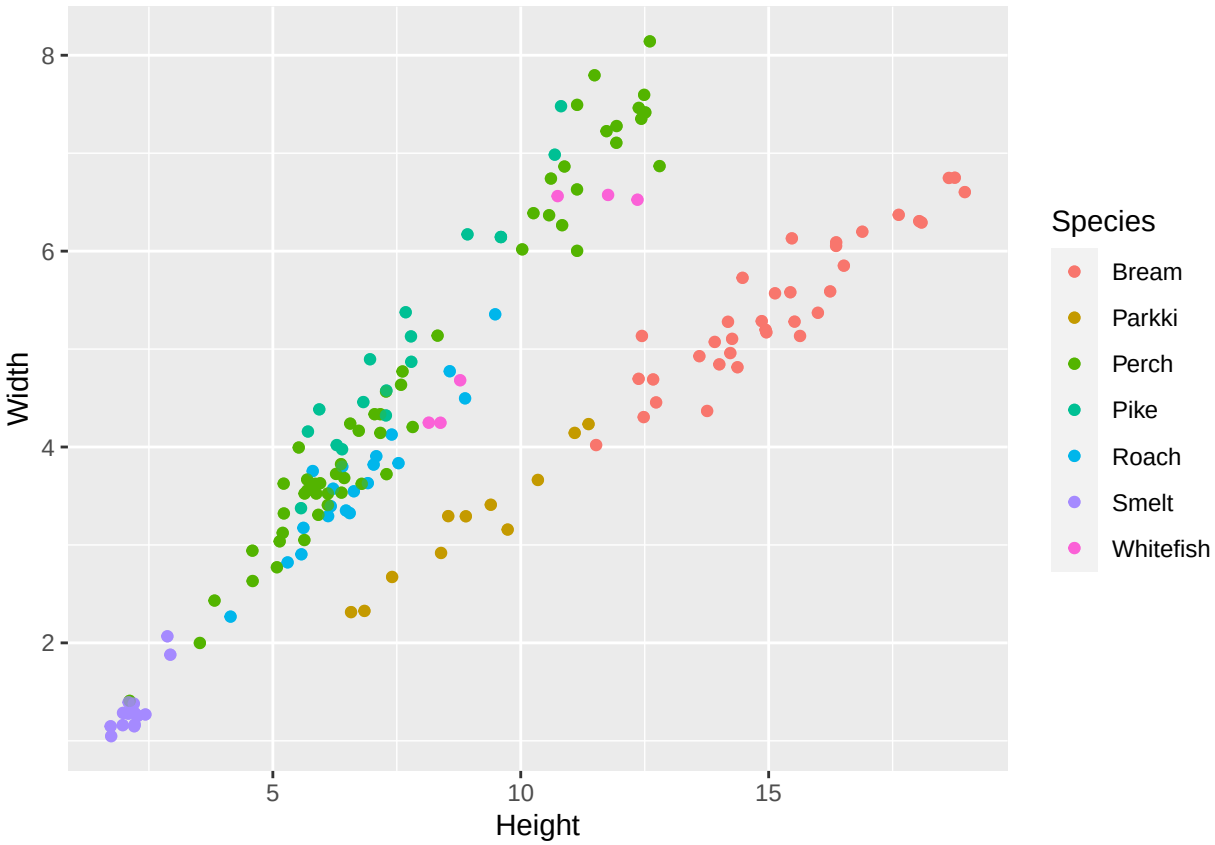

```
ggplot(Fish, aes(x = Weight, y = Width, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```



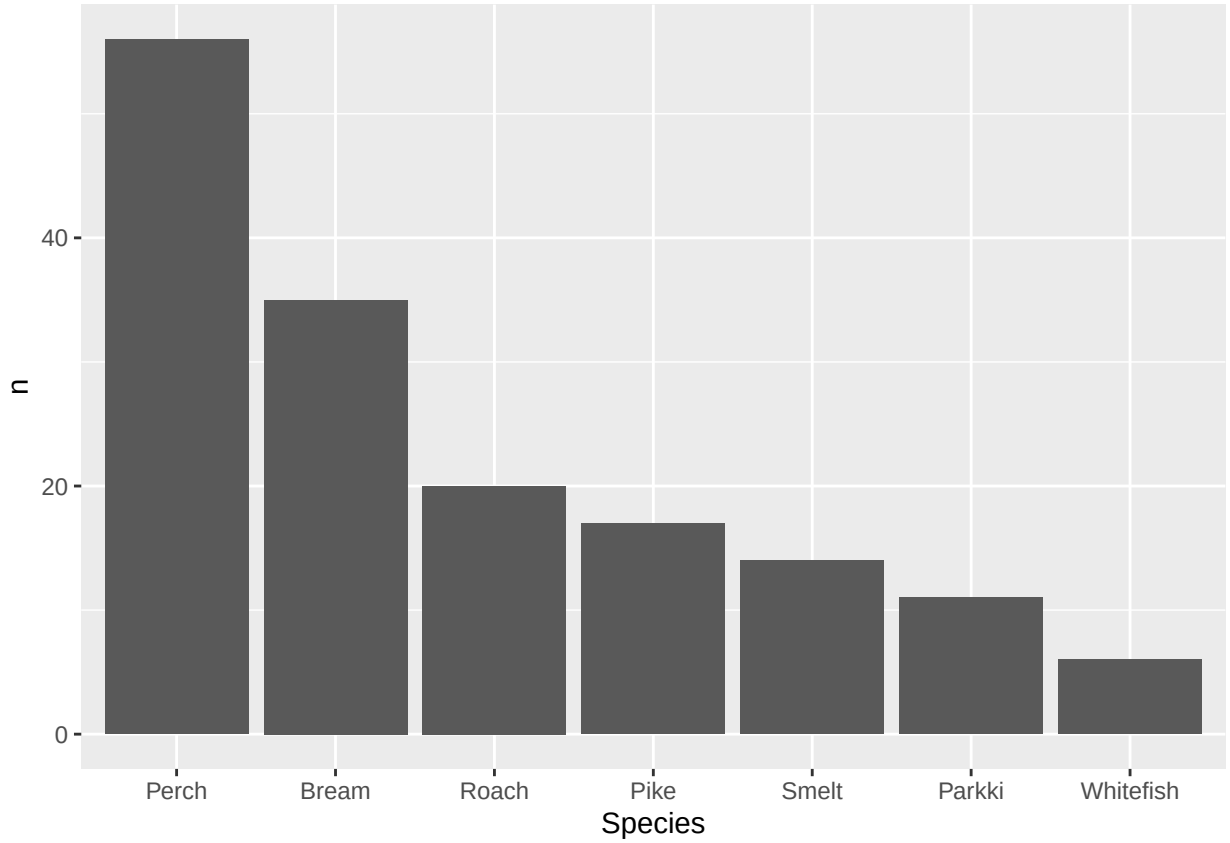

```
ggplot(Fish, aes(x = Length1, y = Length3, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```



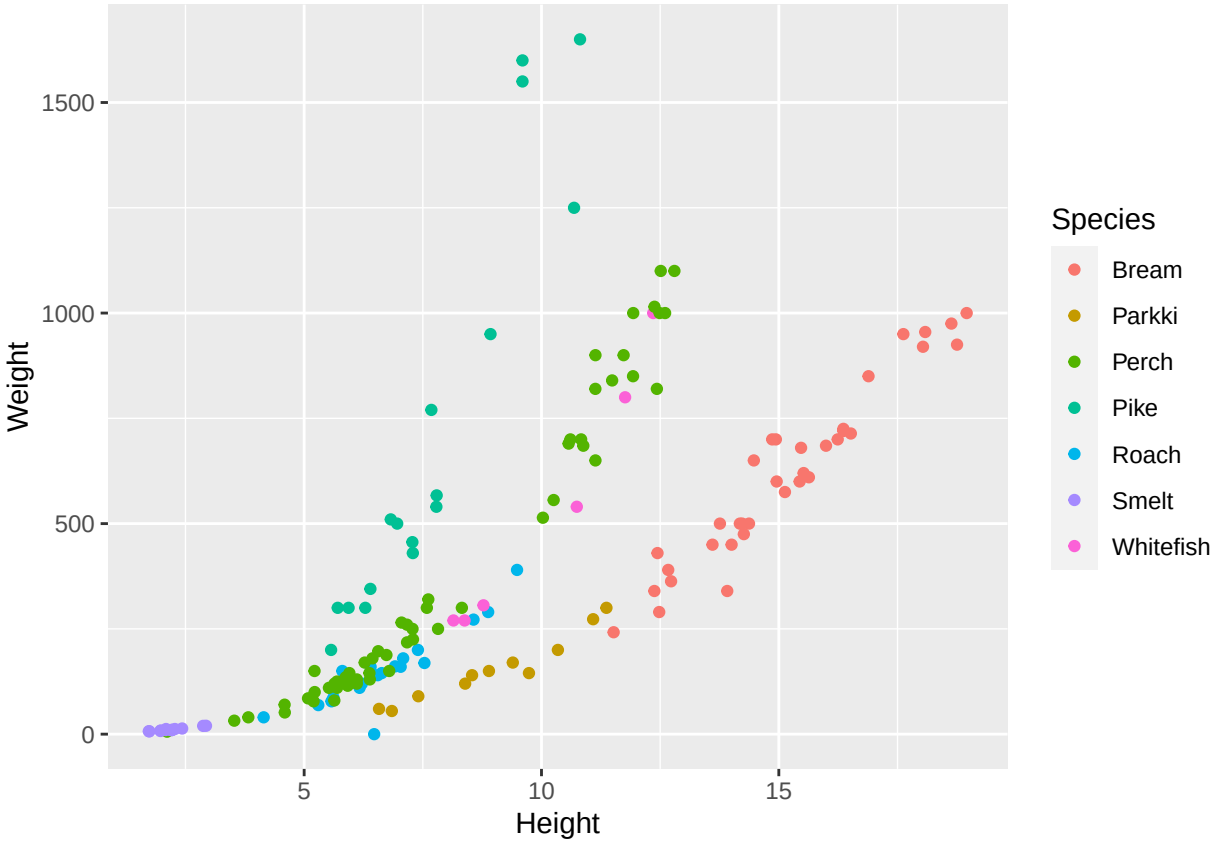

```
ggplot(Fish, aes(x = Height, y = Width, color = Species)) + geom_point() + geom_jitter(alpha = .5)
```




```
Fish %>% count(Species) %>% mutate(Species = fct_reorder(Species, n, .desc = TRUE)) %>%
  ggplot(aes(x = Species, y = n)) + geom_bar(stat = 'identity')
```



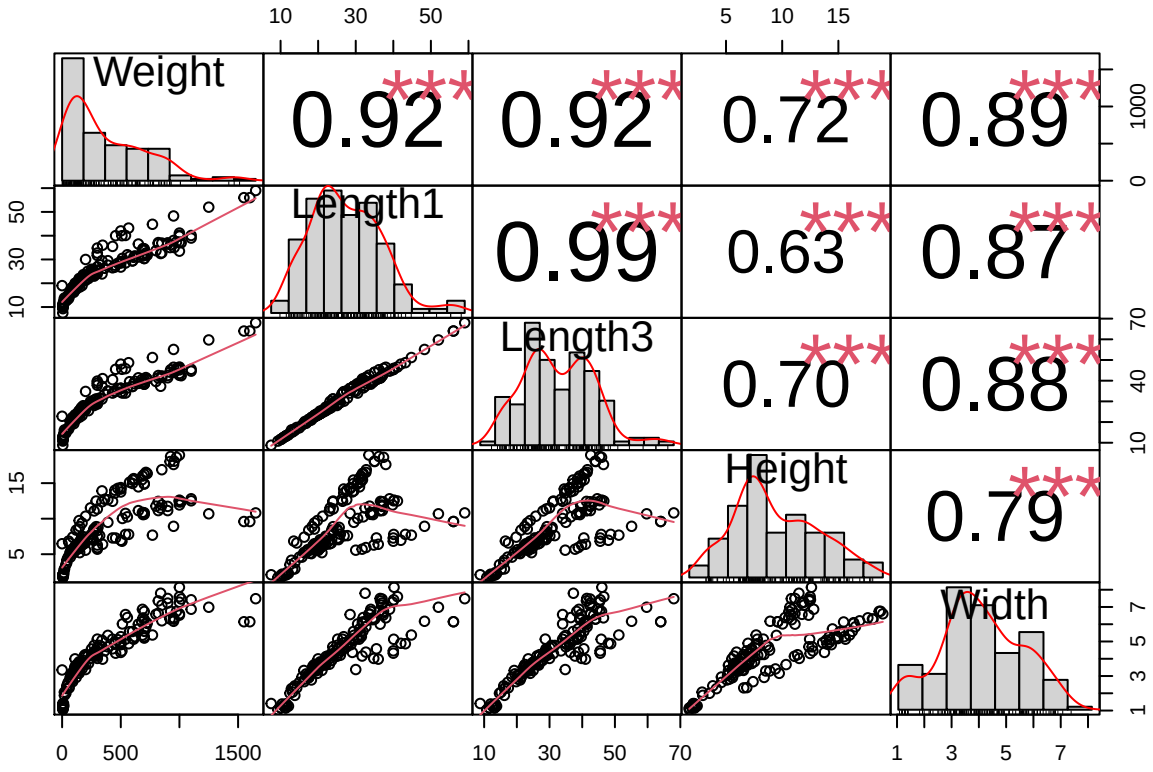

```
ggplot(Fish, aes(x = Height, y = Weight, color = Species)) + geom_point()
```



Our data consists of weight, height, length, and width in grams and cm. This is important because we can visualize this data by creating scatter plots to see how the data works together. We also wanted to create a graph that has a count of the Species in the dataset and we can do this with a barplot. We can see that the variables have a positive correlation meaning that as one variable increases, so should the other. This is important, because we find when implementing a model this is not true for all variables.

Correlation Graph

```
Fish1 <- Fish[, 2:6]
chart.Correlation(Fish1, histogram = TRUE, pch = 19)
```



We wanted to see the correlations of the variables, but typing that over and over is quite tedious. Instead we create a chart that has the correlation between variables. Before we could create this chart we needed to subset the data to only include numeric columns. Weight is most closely related to Length3 with a r = .92 with Height being less correlated at r = .72!

Regression Model

```
set.seed(42)
Fish_split <- initial_split(Fish, prop = .75)
training_data <- training(Fish_split)
testing_data <- testing(Fish_split)
dim(training_data)
```

```
## [1] 120   6
```

```
dim(testing_data)
```

```
## [1] 39  6
```

```
model <- lm(Weight ~ factor(Species) + Length1 + Length3 + Height + Width, data = training_data)
Fish_predict <- predict(model, newdata = testing_data)
error <- Fish_predict - testing_data[["Weight"]]
sqrt(mean(error^2))
```

```
## [1] 90.21256
```

```r
summary(model)
```

```
##
## Call:
## lm(formula = Weight ~ factor(Species) + Length1 + Length3 + Height +
##     Width, data = training_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -231.28  -59.67  -19.16   46.70  429.73
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -1034.651    145.642  -7.104 1.31e-10 ***
## factor(Species)Parkki       258.852     89.480   2.893 0.004611 **
## factor(Species)Perch        258.635    131.023   1.974 0.050915 .
## factor(Species)Pike        -138.774    158.715  -0.874 0.383845
## factor(Species)Roach        148.169    110.065   1.346 0.181031
## factor(Species)Smelt        525.319    141.755   3.706 0.000333 ***
## factor(Species)Whitefish    188.090    109.158   1.723 0.087707 .
## Length1                     -43.444     29.078  -1.494 0.138051
## Length3                      77.386     26.750   2.893 0.004610 **
## Height                        1.866     15.767   0.118 0.906002
## Width                        -4.919     28.655  -0.172 0.864023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 97 on 109 degrees of freedom
## Multiple R-squared:  0.9395, Adjusted R-squared:  0.9339
## F-statistic: 169.3 on 10 and 109 DF,  p-value: < 2.2e-16
```

The goal of this project was to predict fish weight with the variables we have present in the dataset. We began our model by splitting the dataset into a train set and a test set. We next create a model using our training data to see how it will perform on unseen data (test set). Once complete, we get the error and square root it to get the root mean square error of our model.

For those who are not familiar with regression models we can interpret it as such: For every cm increase in Length3, Weight increase by 77.386 grams. For example, let's say you want to predict the weight of a Pike fish and you only have information about the fishes Length3(cross). You can use the formula **predicted_weight = -1034.651 + 77.386(Length3)** created by R to get those results.

Cross Validation

```r
model2 <- train(Weight ~ factor(Species) + Length1 + Length3 + Height + Width, data = Fish, method = "lm
```

```
## + Fold01: intercept=TRUE
## - Fold01: intercept=TRUE
## + Fold02: intercept=TRUE
## - Fold02: intercept=TRUE
## + Fold03: intercept=TRUE
## - Fold03: intercept=TRUE
## + Fold04: intercept=TRUE
## - Fold04: intercept=TRUE
```

```
## + Fold05: intercept=TRUE
## - Fold05: intercept=TRUE
## + Fold06: intercept=TRUE
## - Fold06: intercept=TRUE
## + Fold07: intercept=TRUE
## - Fold07: intercept=TRUE
## + Fold08: intercept=TRUE
## - Fold08: intercept=TRUE
## + Fold09: intercept=TRUE
## - Fold09: intercept=TRUE
## + Fold10: intercept=TRUE
## - Fold10: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```
print(model2)
```

```
## Linear Regression
##
## 159 samples
##   5 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 143, 143, 143, 144, 143, 143, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   95.03971  0.934023  72.4633
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
Fish_predict_final <- predict(model2, Fish)
Fish_predict_dataframe <- data.frame(wgt_predicted = predict(model2, Fish))
print(Fish_predict_dataframe)
```

```
##     wgt_predicted
## 1     281.5592268
## 2     339.5394456
## 3     333.7191599
## 4     417.0025709
## 5     438.4386441
## 6     483.3779870
## 7     472.2912439
## 8     471.9851273
## 9     486.5893097
## 10    530.6572602
## 11    533.4960888
## 12    525.7209188
## 13    523.9074458
## 14    567.9601882
## 15    570.9751650
## 16    572.2199469
```

```
## 17    609.7070329
## 18    623.8382895
## 19    618.8841202
## 20    611.8665898
## 21    659.8970768
## 22    643.3767341
## 23    670.4788353
## 24    715.7729804
## 25    713.4588434
## 26    741.7757507
## 27    715.2362666
## 28    753.8704324
## 29    758.0091257
## 30    812.7296270
## 31    858.0206025
## 32    851.7881722
## 33    900.5609385
## 34    900.0849719
## 35    915.2038813
## 36   -183.5188570
## 37    -25.7685642
## 38      2.3232177
## 39     44.4404613
## 40     33.3687767
## 41     62.6253230
## 42     76.9524197
## 43    106.8825596
## 44    135.9814202
## 45    112.6562090
## 46    180.0591297
## 47    143.0008456
## 48    136.5215138
## 49    259.5999253
## 50    223.4770010
## 51    227.6297544
## 52    250.1228835
## 53    331.8215102
## 54    388.2130419
## 55    534.6114286
## 56    325.4198970
## 57    347.1253391
## 58    399.1431367
## 59    519.5065041
## 60    725.5491582
## 61    869.2559650
## 62    -99.0188028
## 63    -67.4997808
## 64     29.2058061
## 65     94.2546523
## 66    138.9324651
## 67    169.7840576
## 68    174.8467770
## 69    211.4800412
## 70    279.4962368
```

```
## 71    368.4182193
## 72    403.1003284
## 73   -414.0172818
## 74   -181.7009343
## 75   -138.5345037
## 76    -94.5207915
## 77    -33.0651105
## 78     -0.8487059
## 79     -6.5362074
## 80     36.3939198
## 81     53.1682571
## 82     54.1392735
## 83    127.3326305
## 84    129.9587982
## 85    126.8067163
## 86    141.9283758
## 87    164.0228207
## 88    160.4046279
## 89    163.4735326
## 90    161.9390802
## 91    157.4402492
## 92    177.9894921
## 93    184.1011831
## 94    185.2510303
## 95    190.4548108
## 96    213.5277907
## 97    236.8215356
## 98    230.3505482
## 99    257.6691393
## 100   264.7077464
## 101   279.5415220
## 102   300.8268038
## 103   343.4140301
## 104   346.2726339
## 105   345.5178139
## 106   345.9466418
## 107   367.7864997
## 108   377.6890578
## 109   446.8388912
## 110   526.7980918
## 111   642.5297747
## 112   680.4482750
## 113   743.9631914
## 114   696.5159690
## 115   756.4953564
## 116   744.4130971
## 117   819.2694074
## 118   826.2367183
## 119   818.5008901
## 120   872.5939221
## 121   880.6769952
## 122   877.2098297
## 123   876.2940265
## 124   962.4879243
```

```
## 125   968.3191065
## 126   981.4354750
## 127  1008.3961187
## 128  1020.3239136
## 129   202.3274333
## 130   342.0381605
## 131   375.7470073
## 132   376.9513636
## 133   404.1811281
## 134   417.7834962
## 135   588.0728515
## 136   584.4637814
## 137   604.1448983
## 138   682.8856209
## 139   695.3351315
## 140   804.9390733
## 141   952.2470881
## 142  1142.5132005
## 143  1292.2698431
## 144  1292.2698431
## 145  1459.8300795
## 146   -74.0545041
## 147   -43.1582387
## 148   -47.9739855
## 149   -29.4142434
## 150   -13.0810216
## 151    -3.8697473
## 152    14.3130988
## 153    14.8642569
## 154    18.2114904
## 155    26.2551376
## 156    29.1031517
## 157    34.8688320
## 158    91.3885139
## 159   139.0472594
```

What we did above was cool, but it was manual and only one test/train split. What if we wanted to do this multiple times on our dataset. Well we can use the caret package to make this much easier and it also automatically calculates RMSE and MAE.

After creating the cross validation model (10 x 10) we predict it onto the entire dataset. To get those results we have to make sure we create a data frame so we can see the difference in predicted weight and actual weight of the fish.

Conclusion This project attempts to predict Fish weight with the variables provided in the dataset. We stated by importing the dataset, exploratory data analysis, correlations, graphs, and finished with modeling. I hope you enjoy, any comments are greatly apperciated, and try it for yourself and see what results you obtain. *Jeremiah Perkins*